PREDICTIVE MAINTENANCE FOR THE COOLING SYSTEM OF A BUS USING UNSUPERVISED TIME SERIES CLUSTERING

Gautam Vira, Tet Yeap, and Iluju Kiringa
School of Electrical Engineering and Computer Science, University of Ottawa, Ontario, Canada
gvira022@uottawa.ca
tyeap@uottawa.ca
iluju.kiringa@uottawa.ca

ABSTRACT

The widespread incorporation of Internet of Things (IoT) devices in various devices such as mobile phones, vehicles, and security systems is used to collect data. These large volumes of data can be used to train models to make predictions about various things. One such application that uses data from IoT devices is called Predictive Maintenance. Predictive maintenance involves collecting data from machines and using algorithms to analyze the machine's condition or determine if the machine requires maintenance or repairs. The work in this paper presents an approach to perform predictive maintenance for the cooling system of a bus. We also generate a synthetic time series dataset to simulate normal buses and buses with underlying faults and gradual deterioration that would not be detected by the systems. The data is used to train clustering models that identify the deterioration patterns and detect the faults before they result in failure or breakdowns.

Keywords

Internet of Things, Predictive Maintenance, Rotations per Minute (rpm), Dynamic Time Warping (DTW)

1. INTRODUCTION

The widespread incorporation of Internet of Things (IoT) devices in various devices such as mobile phones, vehicles, and security systems to gather data has led to Industry 4.0 [1]. These large volumes of data, known as big data, prove to be very useful in gathering valuable insights about the user's experience and the device or component's performance. One such artificial intelligence application that uses such data has gained popularity and is being used by various organizations is called Predictive Maintenance. Predictive maintenance uses IoT devices embedded in different kinds of devices and transport systems - Road vehicles, Railways, and Aeroplanes - to collect data about their performance and use algorithms and methods such as active databases (rule-based), mathematical models, machine learning, and deep learning to analyze the machine's condition and predict values such as the remaining useful life (RUL) [2] of the machine or determine if the machine requires maintenance or repairs, that is, it predicts the deterioration of a machine or a group of machines that form a system. Prior to such a predictive approach, organizations would have to conduct scheduled check-ups for their vehicles to find out if they needed any repairs or maintenance; however, the scheduled check-ups resulted in downtimes and shut-downs and so this proved to be costly and time-inefficient as many of the vehicles would not require any kind of maintenance or repairs. Furthermore, unexpected breakdowns of vehicles resulted in the organizations incurring heavy costs. Such unexpected breakdowns are common in situations that do not conduct periodic checks to save resources and time.

There are several maintenance strategies employed by organizations such as the aforementioned ones - they can be summarized into three primary categories:

- Corrective Maintenance: This is a reactive measure in which a component or system has suffered failure and needs to be urgently repaired or replaced. Corrective Maintenance occurs as a result of unexpected and unscheduled events.
- Preventive Maintenance: This is a measure that most organizations have in place to maintain the health and ensure the performance of the machinery. This measure involves conducting periodic checks to make sure that nothing is wrong with the equipment, and if a fault is found, maintenance activities are conducted accordingly.
- Predictive Maintenance: This is a measure that estimates and predicts the probability of a component or equipment failing given a future time. Such an approach allows organizations to avoid unexpected failures and skip non-required periodic check-ups that could be very costly.

In this paper, we provide an approach to identify the gradual deterioration (or faults) of the cooling system of a public bus. These kinds of faults are often not noticeable by the sensors on the vehicle nor visible to the eye until the deterioration progresses to a severe level resulting in either component failure or immediate need for maintenance. The data used in this paper was collected in [3] from a public bus of the Société de transport de l'Outaouais (STO) transit service of the Ottawa-Gatineau region. The data - sensor readings relevant to the cooling system of a bus - from a single bus is used to generate synthetic data to simulate approximately 40 buses using Gretel Synthetic's DGAN model [4] and then manually introducing trends that were not captured by the generative model. After generating synthetic data, gradual deterioration in components is introduced to approximately twenty-five percent of the buses. The faults introduced are based on common knowledge and heuristics. Finally, the data is used to train clustering models to identify the sensor readings with deterioration or faults. This way, individual models that identify abnormal sensor readings from the same bus can be grouped to identify the potential fault. The main contributions of this paper are as follows:

- A dataset containing synthetic sensor readings to simulate approximately 40 buses along with additional test cases. Some of these buses have abnormalities (gradual faults) introduced. The dataset can be used to test machine learning algorithms - supervised and unsupervised for similar tasks.
- A clustering approach to conduct predictive maintenance for the cooling system of a public bus. Such an approach could be extended to several other vehicle groups and vehicle subsystems.

This paper is organized as follows. Section 2 expresses the motivation behind choosing the problem of maintenance and its importance, section 3 provides a review of the main works related to conducting predictive maintenance across three different modes of transport - road vehicles, railways, and aircraft. In Section 4, we describe the processes of data generation and processing as well as fault introduction to get realistic data for machine learning applications. In Section 5, we describe the approach taken to conduct predictive maintenance and present the training of clustering models along with the results from the predictive maintenance approach. Finally, Section 6 concludes this paper and offers some interesting directions for future research.

2. MOTIVATION

Most organizations contain preventive measures in place such as conducting periodic check- ups [5] to avoid unexpected failure of equipment. However, most scheduled check-ups result in no fault being found and prove to be unnecessary. A periodic maintenance approach such as this can prove to be inefficient and costly but it is still required to ensure the health status of vehicles. Not doing so would potentially result in unexpected failures of equipment causing downtime and shut-downs. Such critical failures can prove to be extremely costly and must be avoided as much as possible. A way of contemplating the costs associated would be to imagine a car breaking down abruptly, this would be followed by having the car towed, diagnosing the vehicle for faults, and performing maintenance activities such as repairs and/or replacements. If the same event is scaled to the extent of an aircraft breaking down, or to an organization that owns a fleet of aircraft. The monetary costs associated with such a fleet system would be significant, in addition to this cost, and most importantly, the safety of passengers is of the highest priority and unexpected failures could put them in harm's way and such loss is immeasurable. Predictive maintenance is an approach that aims to solve these problems, it seeks to avoid requiring period check-ups and as well as seeks to notify users of potential failures so that maintenance or repairs can be conducted in advance.

The work in this paper aims exactly to do that, save resources, ensure safety, and increase efficiency, by providing an approach for detecting potential faults and failures that can isolate the cause of deterioration to a specific component or component group so that the diagnostic process can be easier.

3. RELATED WORKS

The concept of predictive maintenance has been around for a long time, however, the recent surge in available data and machine learning algorithms has increased its popularity. The main works are covered across three popular modes of transport - road vehicles, railways, and aircraft. Several approaches are used to conduct predictive maintenance in these modes of transport such as using mathematical models, rule-based approaches, machine learning, deep learning, or a combination of these.

The work in [6] uses a rule-based approach to carry out the task of IoT-based predictive maintenance, and the works in [3], [7] use a rule-based approach for the task of deviation detection in a fleet - either on real data or on simulations. The rule-based approach methods are not suitable for detecting gradual faults as noise in the data could easily cross the threshold but that does not indicate a potential fault, additionally, the data values need to cross the threshold to determine a fault as opposed to the trends in the data being captured to observe slow degradation of components.

The authors in [3] use the mathematical model - ICOSMO - for the task of sensor selection to be later on used as part of the algorithm. The work in [8] uses Multi-Target Probability Estimation algorithms to predict the probability and time of failure with the objective of notifying the user a week or two prior to any failure. Mathematical approaches such as maintenance scheduling, reliability estimation, and data selection are useful when included as part of the predictive maintenance algorithm for secondary tasks; relying on mathematical models solely for the task of degradation detection would not be feasible due to their sensitivity to noise and the range of values it was modeled for.

The availability of real data allows for deep learning methods to be employed. The work in [9] used geographic data along with historic maintenance records to train a deep neural network along with autoencoders for the task of time-between-failures (TBF) prediction, which is the time for a machine between two consecutive breakdowns. Deep learning approaches require large volumes of labeled data that are not easily accessible by the public; hence, relying on synthetic data becomes inevitable as well as the need to use traditional machine learning algorithms.

Lastly, the authors in [10] use real sensor data to train supervised machine learning models such as decision trees, random forest, support vector machines, and neural networks to predict the clogging status of the oxygen sensor in the engine. The data used in this approach was collected in a controlled environment by exercising the engine and was then labeled by domain experts, such a process is often not feasible for most organizations.

Similar approaches have been used to conduct predictive maintenance in railways, however, the popular components in railways are the tracks, the wheel bearings, and the pantograph as these are considered to be the most prominent causes of failures in railways [11]. The authors in [12] propose a predictive maintenance approach for railway systems specifically related to point systems. The work uses an Unobserved Components (UC) model set-up in a state space framework and a threshold-based approach to determine if the condition of the point mechanism is good or bad depending on how far the input is from the reference curve; the threshold values are set from past experiences. The work in [13] aims at optimizing the scheduling of maintenance in railways by accounting for unexpected events and uncertainties in the approach; they use mixed integer linear programming to perform maintenance scheduling and use a threshold-based approach for degradation evaluation. The work in [11] proposes a com- plex fuzzy system-based approach using thermal images of two components of the railway systems - the railway track and the pantograph catenary system - to determine the health of the track and rail.

The authors in [14] adopt a rule-based approach to predict the remaining useful life (RUL) by deciding temperature ranges that could negatively impact the wheel bearings, they also consider factors such as the severity of impact and the performance of lubricants given the temperature ranges. The work also included training a neural network to predict the temperature variations of the wheel bearings; the authors had access to past wheel bearing data for the task. The work in [15] proposes an architecture that uses clustering, tree-based algorithms, recurrent neural networks, and reinforcement learning to predict rail and geometry defects and to determine required inspections, the authors used three datasets containing defects, inspection information, and load information. The work in [16] uses large volumes of data - historical detector data, failure records, maintenance records, train information, and weather data - to perform analytics using machine learning. The primary aim of the work is to use machine learning approaches to make accurate predictions and obtain a set of human interpretable rules that can be used by personnel to perform predictive maintenance. In most cases, large volumes of data are not available, also most railways do not have the budget to conduct data collection by deploying the required equipment. Finally, The authors in [17] present the use of simple machine learning techniques for conducting predictive maintenance in railways specifically for railway switches. Their work focuses mainly on employing tree-based machine learning algorithms to make various predictions and they use data collected from a sort of ticket system that is used to generate maintenance requests; the system used for collecting the data was adopted by a railway business.

Another mode of transport that utilizes predictive maintenance is an aircraft. In general, similar techniques can be applied to airplanes or any mode of transport depending on the similarity of data. Approaches relying on historic maintenance data can employ a very similar machine learning model regardless of the mode of transport as long as there is sufficient data to train the model. However, some things to consider are that unlike railways, airplanes and road vehicles have a stochastic aspect due to the manual control by the driver or pilot [18]. The authors in [18] propose an architecture to conduct predictive maintenance in aircraft by using convolutional transformers. In addition to proposing a model for the task, their work also introduces a multivariate time series dataset of many labeled flights consisting of sensor data called the National General Aviation Flight Information Database - Maintenance Classification (NGAFID-MC) dataset. The authors believed that using a deep learning approach was the best way because the data used were time series containing large numbers of time steps. In comparison to using deep learning architectures, traditional machine learning approaches are more popular for the sake of complexity, unavailability of large volumes of data, and simplicity. The work in [19] presents a regression solution for performing predictive maintenance by using only the event logs from post-flight reports. The flight reports consist of event logs that contain information about equipment failure along with some other description such as the system or subsystem that may have caused it. The aim of the regression model is to predict the next occurrence of an event (event of interest or target event) and it does so by determining the risk that the event might occur given the past set of events. The authors in [20] present an approach to use Particle Swarm Optimization (PSO) for model selection and feature selection to perform predictive maintenance by predicting the Remaining Useful Life of aircraft engines. The authors used support vector machines for the task, more specifically a variant of support vector machines - ε-SVR (Support Vector Regression). The primary purpose of this work is to propose an approach that involves the simultaneous tackling of two important machine learning tasks - feature selection and hyperparameter tuning - using Particle Swarm Optimization (PSO) and they used the task of Remaining Useful Life (RUL) prediction to support and evaluate their work. The work in [21] proposes an analysis system that uses clustering, an unsupervised learning technique, to perform predictive maintenance in aircraft, specifically in aero-engines. The approach is similar to the one presented in this paper, however, [21] primarily follows the idea of anomaly detection by finding points that are away from the dense region, as opposed to detecting specific potential faults in individual components; some aero-engines would have to be on the verge of failure for them to be marked as an anomaly in a large fleet of aircraft. Furthermore, they only use Euclidean distance as the metric for clustering which may not give accurate results for data with noise.

Some unique approaches involve using statistical models for wear-and-tear analysis. The work in [22] involves a method to identify a wear model (from a multiple model method) for a component that is degrading non-linearly or without a specific interval time frame by the use of an Extended Kalman Filter. The aim of this work is to optimize the maintenance planning of the systems based on their wear profiles and Remaining Useful Life (RUL) estimates. Lastly, authors in [23] propose an approach that uses proportional hazard models (PHM) for the reliability estimation of components of aircraft by identifying operational factors that impact those components. Two types of Proportional Hazard Models are implemented: time-dependent and time-independent which use the operational attributes as covariates.

4. DATA

The data used in this paper was first collected in [3]. The data consists of sensor readings from a bus of the Société de transport de l'Outaouais (STO) transit service of the Ottawa-Gatineau region. The sensor readings from the bus were received as packets via the SAE J1939 protocol [24] which were then decoded. The messages were transmitted and recorded at an interval of 1 millisecond to 100 milliseconds approximately; the data was collected from a single bus over a duration of approximately ten hours. J1939 messages contain two components: Controller Area Network (CAN) ID that contains the Parameter Group Number (PGN) and the data bytes that contain the Suspect Parameter Number (SPN). The Suspect Parameter Number serves as an identifier of the sensor readings that are encoded in the data bytes and they are grouped together based on the Parameter Group Number based on the component groups. The PGN are identified by their fixed location in the CAN ID and the individual sensor readings belonging to that particular PGN can be identified based on the SPN information of that sensor. Upon locating the relevant bytes for a sensor (or SPN), the values need to be converted from HEX to decimal and scaled using the offset and unit-per-bit values. The J1939 specification document contains all the information that is required to retrieve and decode the physical data values of sensors.

The sensors chosen for this paper belong to the cooling system of the bus. The Parameter Group Numbers found to be the most relevant are PGN 64817 (Fan Drive #2), PGN 65262 (Engine Temperature 1), and PGN 65263 (Engine Fluid Level/Pressure 1). Furthermore, the specific Suspect Parameter Numbers selected are SPN 1598 in PGN 64817 (Fan 2 Speed in rotations-per-minute), SPN 110 in PGN 65262 (Engine Coolant Temperature in degree Celsius), and SPN

111 in PGN 65263 (Engine Coolant Level 1 in percentage). There are other relevant PGN-SPN groups to a bus' cooling system such as Fan 1 Speed (PGN-SPN 65213-1639), Engine Fuel 1 Temperature 1 (PGN-SPN 65262-174), Engine Oil Temperature 1 (PGN-SPN 65262-175), Engine Oil Level (PGN-SPN 65263-98), Engine Oil Pressure (PGN-SPN 65263- 100), and Engine Coolant Pressure 1 (PGN-SPN 65263-109), however, these sensors could not be used because the decoded sensor readings showed that the sensors either were not able to record the values or the stored values were just noise.

4.1. Data Processing and Generation

Level (PGN Three sensor readings were chosen for this work - SPN 1598 in PGN 64817 (Fan 2 Speed in rotations-per-minute), SPN 110 in PGN 65262 (Engine Coolant Temperature in degree Celsius), and SPN 111 in PGN 65263 (Engine Coolant Level 1 in percentage) - as these were the only data bytes that contained values which seemed realistic and accurate. Upon decoding the values, the resultant dataset had the format - elapsedTime, PGN, SPN, decodedValue. The first column (elapsedTime) was the time column in milliseconds since the start of the bus, columns PGN and SPN were categorical columns that contained the Parameter Group Number and Suspect Parameter Number information, and the decodedValue was the continuous column that contained the physical sensor readings for that particular PGN-SPN group. In order to generate synthetic data using a generative model with a long short-term memory network as the generator, categorical features had to be removed and the dataset needed to be restricted and reshaped. The dataset was reshaped to be of the format [Fan Speed, Coolant Level, Coolant Temperature] as columns for the three sensors chosen. Reshaping the dataset resulted in only one sensor reading being in one row of the dataset (example, 4096.0, NaN, NaN), this is because decoded values were appended to the dataset row-by-row to keep the reshaped dataset in line with the time column. The null values were filled using the forward fill method in which an initial value is used to replace the following null values in a column until a non-null value is reached. All these steps resulted in a final time series dataset that contained three sensor readings, one in each column.

The setup of the original dataset containing physical values from three sensors was ready after the reshaping; this dataset was used to generate synthetic data to simulate approximately forty buses. The first step in the generation of synthetic data included using the DGAN model from Gretel [4]. The DGAN model is based on a Generative Adversarial Network (GAN) that uses an LSTM as a generator. A few things to note about the model is that it does not support categorical features as mentioned previously so the dataset had to be restructured so that each column denotes a sensor and contains continuous values (decoded sensor readings). Secondly, the model only allows for a fixed length of sequences, and variable-length sequences were not supported. An example to explain the sequence length would be to consider a dataset that contains the temperature readings of a room every hour for seven days; such a dataset could contain a sequence length of twenty-four resulting in seven sequences of daily temperature readings. The data available for use in this work was only sensor data for a single bus so splitting the dataset into a fixed sequence length required repeated trying and testing of different values. Several combinations of hyperparameters were tried and tested to generate realistic data that also captured the temporal relations as well as included some randomness to simulate the stochastic aspects that lead to certain trends in the sensor readings. The hyperparameters configured were - the number of layers in the generator and discriminator, the number of units in an LSTM cell, the sequence length, the batch size, and the number of epochs. Finally, it was found that smaller batches and sequence lengths along with three or four layers in the network were found to be the best combination. However, the resultant synthetic data failed to capture almost all of the temporal relations. Figure 1 contains the sensor readings of the coolant temperature and a comparison between the real data and the synthetic data. The values are plotted using a moving average method using a window size of one thousand; this is to enhance visibility and interoperability. It is visible in Figure 1 that the two lines representing real (in red) and synthetic (in blue) data are not similar. The readings represented by the blue line are almost flat and constantly around ninety degrees Celsius in comparison to the red line which contains temperature dips. Another instance of this was visible in the fan speed readings, the real data contained only two unique values - 4096 rpm and 0 rpm - with only fifty readings that had a value of 0 displaying that the fan was turned off for a few seconds across the whole time period and the fan's operational value was at 4096 rpm. The synthetic data had no values of 0 rpm but consisted of values around 4096 rpm which does not seem to be accurate for the functionality of an electric fan. One benefit of synthetic data generation was observed in the values of coolant level. The real sensor readings consisted of two values - 50% and 0; this indicates that the accurate value of the coolant level was 50% of the total container and there were a lot of readings in which the sensor was not able to record the real value so it was recorded as 0%. On the other hand, the synthetic data only had the actual value which was 50% but for all the generated buses which does not hold true in real life, that is, buses could operate with varying levels of coolant without it causing a problem or failure.

The main reason behind using the generative model to generate synthetic data was to get as close to the distributions and trends as possible. Although many temporal relations have not been captured, the generated data provides some useful insights such as there was no relation found between the fan turning off and the coolant temperature and levels, the coolant temperature's normal working temperature is around ninety degrees Celsius and the temperature dips needed an explanation because the bus was running for almost the whole duration, and lastly, the coolant level values included 0% but this should be disregarded and filtered out. The aforementioned insights will be explained later in the paper because they are relevant to simulating the trends that were not captured by the GAN model.

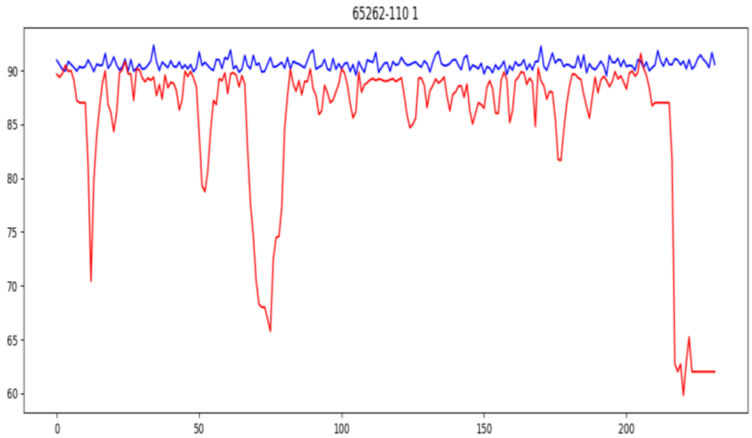

Fig. 1. Coolant temperature comparison between real (red) and synthetic data (blue)

4.2. Manual Trend Introduction

In addition to the uncaptured trends, the synthetic data generated using the DGAN model contained a lot of fluctuations from one reading to the next which made it appear very noisy. The problem of noise was tackled in the data preprocessing step. To make the synthetic data more realistic, the temporal trends mentioned earlier that were not captured by the generative model needed to be manually introduced. An initial step made sure that all the data generated was in the permissible data ranges of the particular sensors, that is, the synthetic sensor readings could not have a value that is impossible to record due to the fixed units-per-bit so all the data was fit between the minimum and maximum values allowed for each sensor.

All generated buses had a coolant level of 50% which is not true in real life, therefore, the coolant levels of the buses were changed based on probability - there was a 30% chance that the buses' coolant level would be increased by a value between 5 and 15 percent (resulting in the final values being between 55% and 65%) and a 15% chance that the coolant level would be increased by a value between 15% and 30% so the resulting coolant levels would be between 65% and 80%. Figure 2 (Real coolant level in %) and Figure 3 (Synthetic coolant level in %) show the difference between the original data and the generated data.

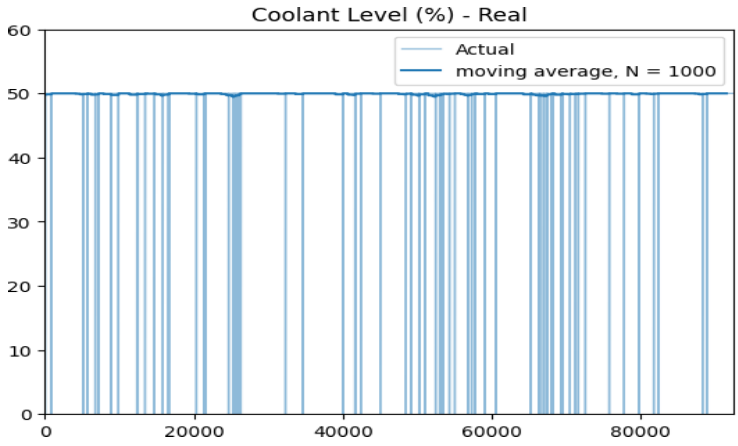

Fig. 2. Real Data Coolant Level

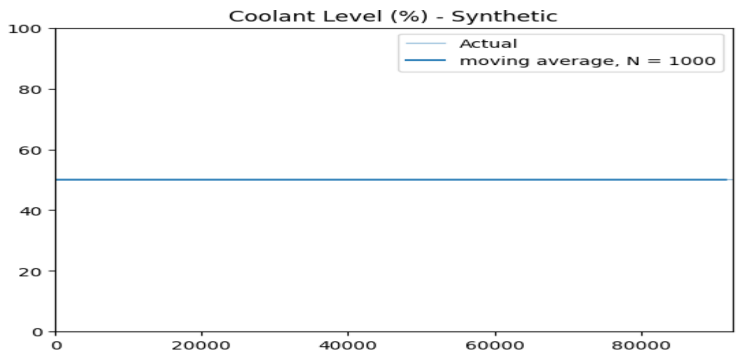

Fig. 3. Synthetic Data Coolant Level

The coolant temperature dips that existed in the original data were missing in the synthetic data. The steep temperature dips from 90 degrees Celsius to approximately 60 degrees Celsius were justified by a few reasons - the engine speed was observed for the same time as the temperature dips and an rpm of zero suggested that the engine was not running whenever there was a large temperature dip, the sensor readings were recorded in the month of November when it is usually cold in Canada, the fall in temperature followed by a rise took place over thirty minutes - these reasons imply that the bus was not running (turned off) for a period of approximately twenty minutes which resulted in the engine cooling down faster due to the cold weather. The introduction of temperature dips was based on the assumption that a bus that was thought to be in perfect condition would have one break causing a significant temperature dip, a significant dip could be caused only in the cold months and not during summer because it would take longer than twenty minutes for the engine to cool down and consequently the coolant, buses would not be turned off immediately after turning them back on (no consecutive temperature dips), and that a bus would not have a temperature dip in the last twenty percent of its up-time due to a break. The dip sizes were either small (indicating that the bus was stationary for a small period of time in the cold), medium (indicating that the bus halted for a few minutes or was turned off for approximately five minutes, or large (indicating that the bus was turned off for a break). The sizes of the dips were determined by the number of time steps for which the temperature will keep decreasing and relatively the number of time steps for it to come back to the functioning temperature. Furthermore, the temperature decline was not consistent or uniform; there was a very small probability that the temperature would go up by one degree, a higher probability that it would go down by one degree, and a significantly high probability that it would remain the same; this resulted in an inconsistent drop which is how sensor recordings appear in real data. However, the temperature was not allowed to go below or above a set threshold based on acceptable coolant temperature values. Similarly, the temperature increase was not uniform and the probabilities of the temperature increasing or decreasing or remaining the same were distributed as mentioned earlier but over fewer time steps. Figure 4 (Real coolant temperature in degrees Celsius) and Figure 5 (Synthetic coolant temperature in degrees Celsius) show the difference between the original data and the generated data.

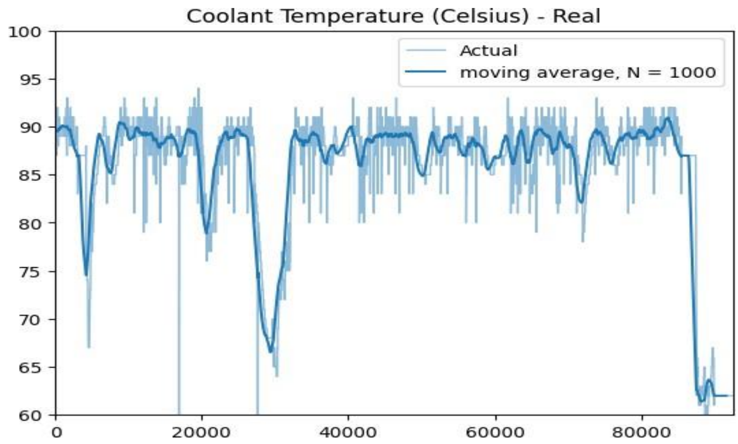

Fig. 4. Real Data Coolant Temperature

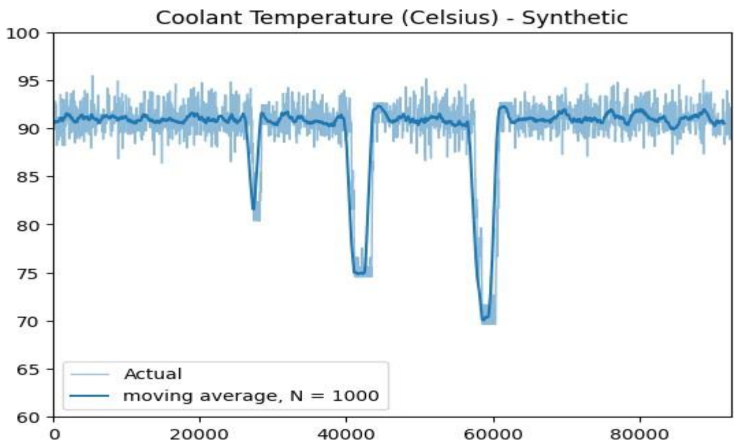

Fig. 5. Synthetic Data Coolant Temperature

The rotations-per-minute (rpm) at which the radiator fan operated was found to be 4096. Firstly, all the values that were not 0 were made equal to the acceptable operation value - 4096 rpm. Secondly, as mentioned earlier, the generative model failed to capture the trend of the fan turning off, that is, the rpm being zero. An effort was made to find out the threshold behind the fan turning off by comparing the coolant temperature and fan speed at the same time steps but there was no relation found between the coolant temperature decreasing and the fan turning off, the fan speed was also compared to the engine speed data but the recorded speed data consisted of just noise so no relation could be established apart from the readings with a value of 0 rpm which implied that the engine was turned off. Therefore, the fan speed was set to zero for a few readings during the temperature drop which seemed to be the most justifiable approach instead of randomly introducing fan turn-offs across the data. Figure 6 (Real fan speed in rpm) and Figure 7 (Synthetic fan speed in rpm) show the difference between the original data and the generated data.

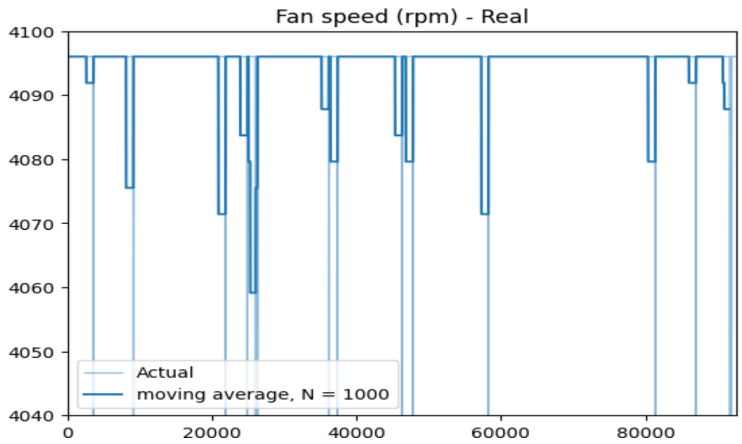

Fig. 6. Real Data Fan Speed

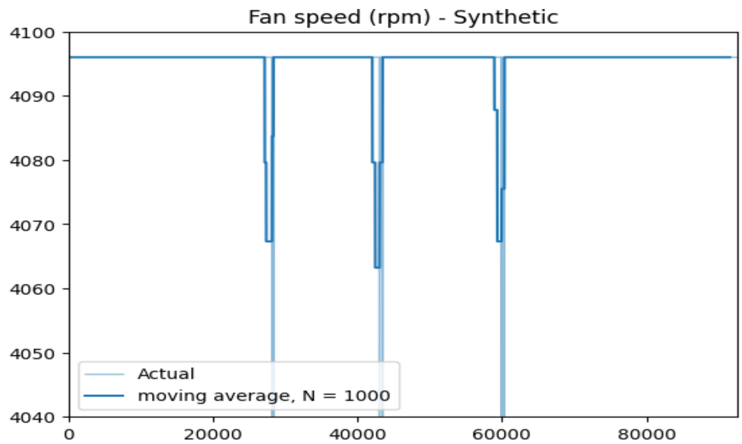

Fig. 7. Synthetic Data Fan Speed

The real data and the generated data with the trends introduced do not appear to look the same but it still seems like a step in the correct direction to generate good-quality simulations as opposed to previous works in which synthetic data was generated by simply fitting the real data to a distribution and sampling from it such as in [3]. Moreover, all the simulated buses have unique trends depending on their coolant levels, temperature dip sizes and frequencies, and fan speed patterns. All the trends were introduced based on probability so there are also some instances in which there are no temperature drops which implies that the bus was operating in summer and was running for the whole duration, consequently, the fan was not turned off for that bus up-time indicating that the fan ran throughout the duration of the bus' journey. The original data had only fifty recordings for which the fan was turned off so the fan turning off was not a very important trend. The trends introduced cover many possibilities and scenarios that, if used in the training of a model in the unavailability of real data, will enable the model to be robust against seasonal and operational factors that are acceptable and normal without them being interpreted as anomalies.

4.3. Fault Introduction

After the trends were introduced to the generated synthetic data, the sensor data of approximately forty buses was simulated. However, the synthetic data so far was assumed to be of normal buses, that is, without any fault. Therefore, the next step was to introduce faults to the existing synthetic data. The faults introduced were gradual and subtle, in essence, they were not abrupt or large enough to trigger caution warnings, failures, or breakdowns in buses because those faults can be easily identified using a threshold-based approach. The assumption for fault introduction was that 25% of the buses in the dataset had potential faults; this assumption was made for simplicity to avoid imbalance problems because of the lack of large volumes of data to tackle such a problem. Hence, eight buses had potential faults in them. The faults introduced were based on heuristics, common sense, and knowledge base. There were four types of faults introduced: The first fault involves lowering the coolant level of two buses by 10% and 15% respectively to indicate a potential leakage in the cooling system. The reduction of the coolant level was linear and over the duration of the buses' up-time, indicating that the coolant was leaking throughout the buses' trips. Figure 8 illustrates the first fault in which the coolant level of a bus was reduced by 15% from its initial value. Note that the reduction was based on 15% of the initial value and not 15% of the entire coolant capacity.

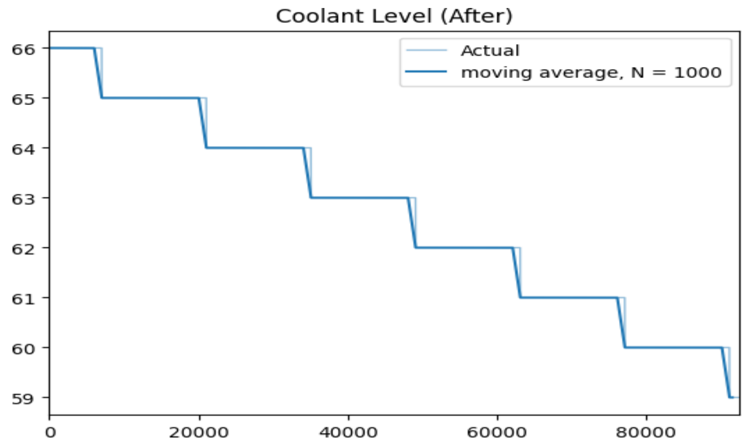

Fig. 8. Coolant level with a 15% decrease

The second fault involves lowering the rpm at which the fan of two buses operated by 30% and 35% respectively and increasing the coolant temperature slightly (by 5% and 7% respectively) to indicate a fan performance deterioration. The decrease in the rpm was uniform and for all the values that were not 0. On the other hand, the increase in the coolant temperature was linear and over the duration of the buses' just like the coolant level decrease. Figure 9 shows the fan speed decrease, it is visible that it is under the ideal rpm of 4096.0. Figure 10 shows the slight increase in the coolant temperature.

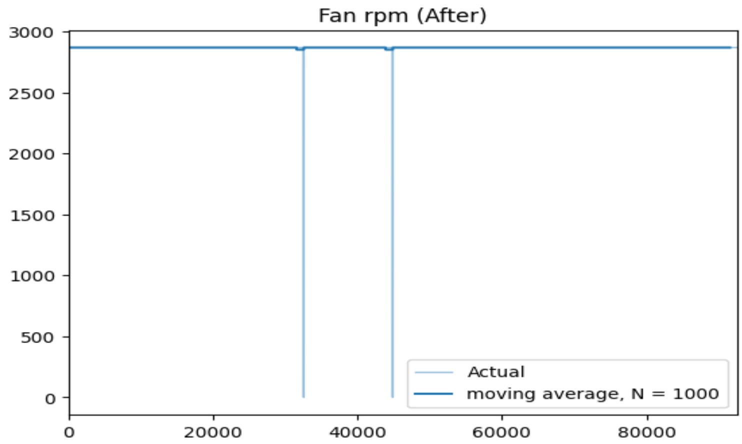

Fig. 9. Fan Speed with a 30% decrease

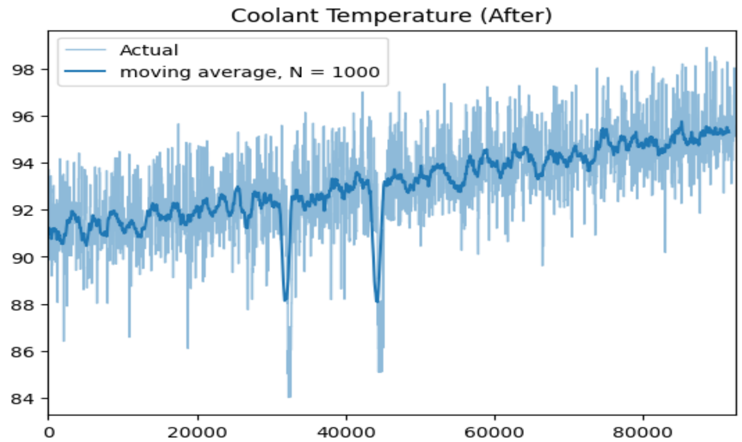

Fig. 10. Coolant Temperature with a 5% increase

The third fault involves increasing the fan rpm of two buses by 10% and 15% to indicate that the fan is rotating above the rated rpm of 4096 and it could cause wear-and-tear faster and lead to failure. Figure 11 displays the fan speed being above the standard rpm of 4096.

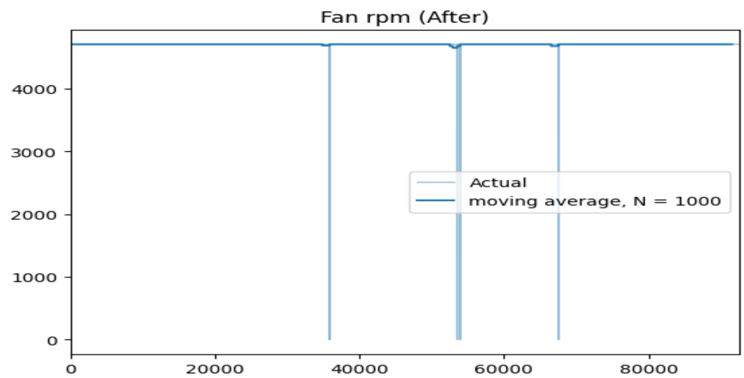

Fig. 11. Fan Speed with a 15% increase

Lastly, the fourth fault involves increasing the coolant temperature by 10% and 15% to indicate a cooling system issue that cannot be caused by the coolant level or fan performance. The purpose of this fault is to detect the temperature change that could have been caused by other cooling components for which the data could not be used in this work. Figure 12 displays the increase in the coolant temperature which is similar but more prominent than in the fault before.

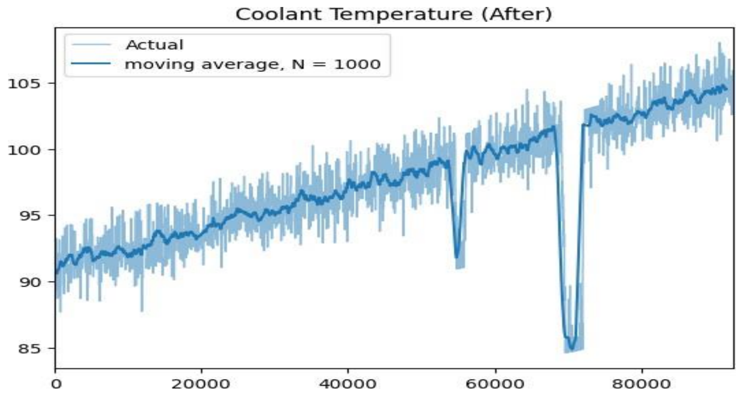

Fig. 12. Coolant Temperature with a 15% increase

5. PREDICTIVE MAINTENANCE APPROACH

K-Means clustering is used for detecting deterioration and unusual trends in the generated data. The metrics used to perform the clustering are Euclidean distance and Dynamic Time Warping (DTW) [25]. The primary idea is to form clusters containing normal buses and clusters that identify component deterioration and faults. K-means clustering using Euclidean distance is used as the baseline to compare to K-Means clustering using dynamic time warping (DTW). The reason behind choosing a clustering approach is mainly the lack of real data with labels. Deep learning approaches have been rather popular amongst other methods due to the increasing computational powers of systems and the possibility of performing heavy computations at the cloud level instead of individual computers embedded in each vehicle. However, in cases of lack of labeled data, traditional machine learning algorithms perform incredibly well. Deep learning methods largely rely on the quality of data and its being labeled, furthermore, there is usually an imbalance in such tasks because most of the vehicles function normally and only a few have potential problems - it is difficult for deep learning methods to tackle such imbalance problems, especially without accurate and sufficient data.

The clustering is performed for each sensor individually to identify any unusual patterns in the component that might not have any relation to the other components. The clustering results for each sensor for a bus can be combined the identify the type of fault and cause of deterioration by experts. The preprocessing step includes converting all the data to integer type for memory efficiency, doing so results in no loss of information because the sensor recordings are whole numbers. The readings in which the coolant level value is 0% are removed as they are noise. The coolant temperature data is noisy and has fluctuations which is visible by the roughness of the line in Figure 4 and Figure 5, so the moving average of the time series was calculated and used for smoothing the data. Additionally, only 10% of coolant level and fan speed readings for each bus were used by sampling at equal intervals; and even fewer readings of the coolant temperature to further make it less noisy. The data for each sensor is then combined and formatted into time series datasets containing approximately 40 buses. The coolant level dataset contained various values that are not relevant to any faults, for example, one bus may have a coolant level of 85% and another with just 50% but that does not mean that there is anything wrong with either of them. Therefore, the shape of the time series was relevant to the problem at hand and not the magnitude of the values. Hence, the coolant level data was scaled to a distribution with zero mean and unit variance, doing so scaled all the normal data to have the same value and trends could be observed between the values of 1 and -1.

Euclidean distance is chosen as the base approach because it is the common choice for clustering due to its complexity and simplicity; it involves calculating the distance between a time series and the cluster centroid point-by-point without taking into account any trends or patterns. However, complex trends such as in the coolant temperature are difficult to distinguish using just a simple metric which is why dynamic time warping is used. Dynamic time warping temporally aligns sequences and calculates the distance between points that are similar even if the indices of the two points do not align in the time series. However, the complexity of DTW is high compared to the Euclidean distance method - $O(n * m)$, where n is the length of sequence A and m is the length of sequence B. Therefore, DTW should be used only if Euclidean distance fails to form accurate clusters.

The choice of K (number of clusters) is based on the types of faults introduced for each sensor. The number of clusters for the coolant level is chosen to be 2, the number of clusters for the fan speed is chosen to be 3, and the number of clusters for the coolant temperature is chosen to be 2. The chosen values of K are also validated by using the Sum of squared distances, which is the sum of the distances of each point in a cluster to the cluster centroid, for K values 2 to 5. The results show that the aforementioned chosen values of K based on the type of faults are optimal because the sum of squared distances does not decrease significantly by increasing the value of K beyond the optimal value. Moreover, the quality of the clusters is determined by using the silhouette score. The silhouette score is calculated: $b - a/max(a, b)$, where a is the mean distance of a point with the other points of its cluster and b is the distance between the single point and its nearest cluster.

The results of the Euclidean clustering and DTW clustering for the coolant level dataset are the same. They both perform incredibly well at identifying the decline in the coolant level so the Euclidean K-Means clustering model can be used for this problem for efficiency. The silhouette score of the K=2 cluster for the coolant level was 0.99 which is extremely accurate. Figure 13 shows the two clusters, one contains all the normal buses and the other contains the two buses with a decline in the coolant level. The thick red line in Figure 13 and the following figures depicts the cluster centroid while the less opaque gray lines depict individual time series.

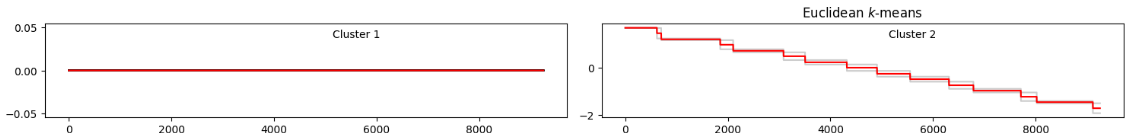

Fig. 13. K=2 Clusters for Coolant Level

Euclidean-based clustering fails to detect the slightly higher rpm in the fan speed dataset. The reason for this might be that the 0 rpm values in one bus might result in a large distance for congruent non-zero points in other buses (normal without faults) so the slight increase does not result in a large enough distance for it to be significant enough to be identified as a separate cluster. Figure 14 shows the clustering using Euclidean, it is visible that cluster 1 includes the buses with the increased fan speed and cluster 3 just contains one time series which does not have any faults. On the other hand, the DTW-based clustering was able to identify the two faults introduced to the fan speed and cluster them correctly with a silhouette score of 0.88. Figure 15 shows the clusters using DTW and it is visible that all the faults are identified and clustered correctly.

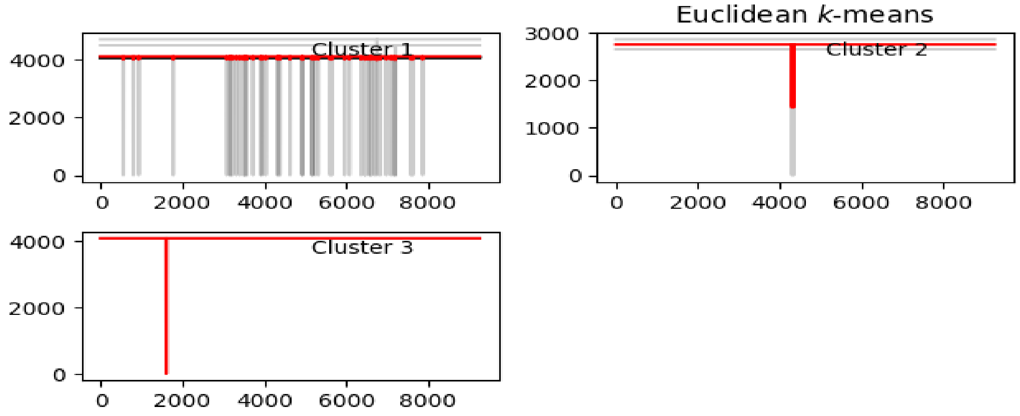

Fig. 14. K=3 Clusters (Euclidean) for Fan Speed

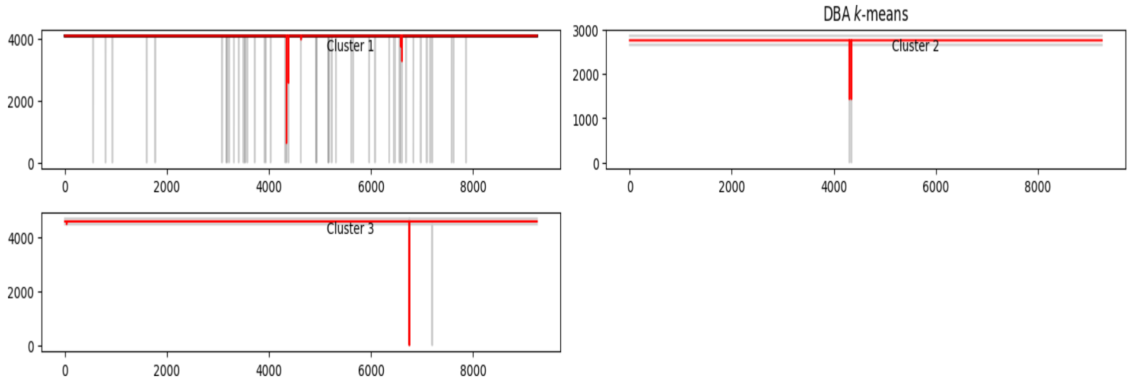

Fig. 15. K=3 Clusters (DTW) for Fan Speed

Clustering for the coolant temperature dataset was only performed using dynamic time warping. This is because the data has complex trends and noise that are not distributed uniformly, it is sure that using just the Euclidean method would not be enough to distinguish the trends from faults and give accurate results. The DTW clustering is able to identify the faults and cluster them appropriately, although, it fails to detect the rise in temperature in one bus. The rise in temperature is very subtle and is not detected by the model. Figure 16 shows the clusters using DTW and it is visible that there is a time series in cluster 2 in which the temperature increases but it is clustered along with the normal buses. The silhouette score of the model is 0.64 which is reasonably high which is evident from the faulty buses being mostly clustered accurately.

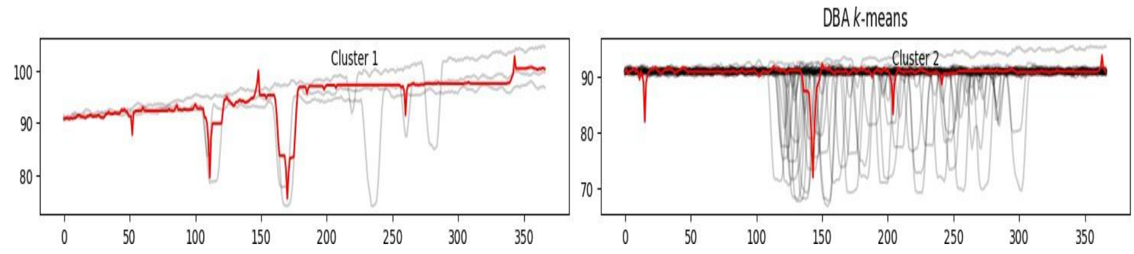

Fig. 16. K=2 Clusters (DTW) for Coolant Temperature

In summary, all the models perform very well and are able to identify the faults and cluster them appropriately. The model for the coolant temperature is unable to identify a single faulty trend but the model for the fan speed is able to identify the decreased rpm in the same which caused the increase in temperature, so when put together, all faults are accurately detected by the models.

6. Conclusion

The work in this paper addresses the issue of common maintenance strategies being costly and inefficient. It shines a light on an alternative strategy that is cost and time efficient - predictive maintenance - and presents an approach to perform it for the cooling system of a bus. In this paper, we also generate a synthetic time series dataset to simulate buses that are normal and buses with underlying faults and gradual deterioration that would not be identified by the systems because the faults have not crossed a threshold after which they result in component failure and damage. The aim of the dataset was to enable the training of a model that could detect these underlying faults accurately and distinguish them from the buses that are normal and without any underlying issues. Doing so could allow the user to take preemptive measures to have the problem rectified without resulting in any failures and breakdowns. Such an approach ensures efficiency in terms of cost, resource, and time and most importantly the safety of the user.

The clustering models in this work were trained mainly using synthetic data and for only three sensors because other sensor data from the cooling system could not be used as it was just noise. A direction forward would be to gather real data from different bus models and in different weather conditions over a period of time. Furthermore, any buses experiencing faults could be labeled and the previous trips of those buses could be analyzed and used in the training of the model to find and predict the trends that resulted in failure. The data and models in this paper belong to the cooling system, however, the approach presented in this paper could be extended to any subsystem of a bus such as the braking system or the engine performance system. Creating models for these subsystems and combining them altogether could be used to perform predictive maintenance for a whole bus so that no faults could go undetected.

Acknowledgements

Acknowledgements go here.

References

[1] N. Davari, B. Veloso, G. de A. Costa, P. M. Pereira, R. P. Ribeiro, and J. Gama, “A Survey on Data- Driven Predictive Maintenance for the Railway Industry,” Sensors, vol. 21, no. 17, p. 5739, Aug. 2021, doi: https://doi.org/10.3390/s21175739.

[2] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, “Remaining useful life estimation – A review on the

[3] statistical data driven approaches,” European Journal of Operational Research, vol. 213, no. 1, pp. 1–14, Aug. 2011, doi: https://doi.org/10.1016/j.ejor.2010.11.018.

[4] P. Killeen, I. Kiringa, and T. Yeap, “Unsupervised Dynamic Sensor Selection for IoT-based Predictive

[5] Maintenance of a Fleet of Public Transport Buses,” ACM Transactions on Internet of Things, Apr. 2022, doi: https://doi.org/10.1145/3530991.

[6] “Gretel DGAN - Gretel.ai,” Gretel.ai, 2022. https://docs.gretel.ai/create-synthetic-data/models/synthetics/gretel-dgan.

[7] J. Xie, J. Huang, C. Zeng, S.-H. Jiang, and N. Podlich, “Systematic Literature Review on Data-Driven Models for Predictive Maintenance of Railway Track: Implications in Geotechnical Engineering,” Geo- sciences, vol. 10, no. 11, p. 425, Oct. 2020, doi: https://doi.org/10.3390/geosciences10110425.

[8] R. Dhall and V. K. Solanki, “An IoT Based Predictive Connected Car Maintenance Approach,” International Journal of Interactive Multimedia and Artificial Intelligence, vol. 4, no. 3, p. 16, 2017, doi: https://doi.org/10.9781/ijimai.2017.433.

[9] S. Byttner, T. Rognvaldsson, and M. Svensson, “Consensus self-organized models for fault detection (COSMO),” Engineering Applications of Artificial Intelligence, vol. 24, no. 5, pp. 833–839, Aug. 2011, doi: https://doi.org/10.1016/j.engappai.2011.03.002.

[10] Sinaiski and Halasya Siva Subramania, “Predictive Maintenance with Multi-target Classification Models,” Lecture Notes in Computer Science, pp. 368–377, Jan. 2010, doi: https://doi.org/10.1007/978-3-642- 12101-2 38.

[11] C. Chen, Y. Liu, X. Sun, C. D. Cairano-Gilfedder, and S. Titmus, “Automobile Maintenance Prediction Using Deep Learning with GIS Data,” Procedia CIRP, vol. 81, pp. 447–452, 2019, doi: https://doi.org/10.1016/j.procir.2019.03.077.

[12] D. Giordano et al., “Data-driven strategies for predictive maintenance: Lesson learned from an automotive use case,” Computers in Industry, vol. 134, p. 103554, Jan. 2022, doi: https://doi.org/10.1016/j.compind.2021.103554.

[13] M. Karakose and O. Yaman, “Complex Fuzzy System Based Predictive Maintenance Approach in Railways,” IEEE Transactions on Industrial Informatics, vol. 16, no. 9, pp. 6023–6032, Sep. 2020, doi: https://doi.org/10.1109/tii.2020.2973231.

[14] D. J. Pedregal, F. P. Garc`ıa, and F. Schmid, “RCM2 predictive maintenance of railway systems based on unobserved components models,” Reliability Engineering & System Safety, vol. 83, no. 1, pp. 103–110, Jan. 2004, doi: https://doi.org/10.1016/j.ress.2003.09.020.

[15] Consilvio, A. D. Febbraro and N. Sacco, ”A Rolling-Horizon Approach for Predictive Maintenance Planning to Reduce the Risk of Rail Service Disruptions,” in IEEE Transactions on Reliability, vol. 70, no. 3, pp. 875-886, Sept. 2021, doi: 10.1109/TR.2020.3007504.

[16] Daniyan, K. Mpofu, M. Oyesola, B. Ramatsetse, and A. Adeodu, “Artificial intelligence for predictive maintenance in the railcar learning factories,” Procedia Manufacturing, vol. 45, pp. 13–18, 2020, doi: https://doi.org/10.1016/j.promfg.2020.04.032.

[17] P. C. Lopes Gerum, A. Altay, and M. Baykal-Gürsoy, “Data-driven predictive maintenance scheduling policies for railways,” Transportation Research Part C: Emerging Technologies, vol. 107, pp. 137–154, Oct. 2019, doi: https://doi.org/10.1016/j.trc.2019.07.020.

[18] H. Li et al., “Improving rail network velocity: A machine learning approach to predictive maintenance,” Transportation Research Part C: Emerging Technologies, vol. 45, pp. 17–26, Aug. 2014, doi: https://doi.org/10.1016/j.trc.2014.04.013.

[19] Z. A. Bukhsh, A. Saeed, I. Stipanovic, and A. G. Doree, “Predictive maintenance using tree-based classification techniques: A case of railway switches,” Transportation Research Part C: Emerging Technologies, vol. 101, pp. 35–54, Apr. 2019, doi: https://doi.org/10.1016/j.trc.2019.02.001.

[20] H. Yang, A. LaBella, and T. Desell, “Predictive Maintenance for General Aviation Using Convolutional Transformers,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 11, pp. 12636–12642, Jun. 2022, doi: https://doi.org/10.1609/aaai.v36i11.21538.

[21] P. Korvesis, S. Besseau and M. Vazirgiannis, ”Predictive Maintenance in Aviation: Failure Prediction from Post-Flight Reports,” 2018 IEEE 34th International Conference on Data Engineering (ICDE), Paris, France, 2018, pp. 1414-1422, doi: 10.1109/ICDE.2018.00160.

[22] El Afia and M. Sarhani, “Particle Swarm Optimization for Model Selection of Aircraft Maintenance Predictive Models,” Mar. 2017, doi: https://doi.org/10.1145/3090354.3090402.

[23] Y. Bao, G. Rui, and S. Zhang, “A Unsupervised Learning System of Aeroengine Predictive Maintenance Based on Cluster Analysis,” Proceedings of the 2020 International Conference on Aviation Safety and Information Technology, Oct. 2020, doi: https://doi.org/10.1145/3434581.3434619.

[24] W. O. L. Vianna and T. Yoneyama, “Predictive Maintenance Optimization for Aircraft Redundant Systems Subjected to Multiple Wear Profiles,” IEEE Systems Journal, vol. 12, no. 2, pp. 1170–1181, Jun. 2018, doi: https://doi.org/10.1109/jsyst.2017.2667232.

[25] W. J. C. Verhagen and L. W. M. De Boer, “Predictive maintenance for aircraft components using propor- tional hazard models,” Journal of Industrial Information Integration, vol. 12, pp. 23–30, Dec. 2018, doi: https://doi.org/10.1016/j.jii.2018.04.004.

[26] “SAE J1939 Standards Collection - SAE J1939 Standards Collection on the Web,” www.sae.org. https://www.sae.org/standards/development/ground-vehicle/sae-j1939-standards-collection-on-the-web.

[27] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (AAAIWS’94), AAAI Press, pp. 359–370, 1994.

Author

Gautam Vira Received his Bachelor’s degree with Honours in Computer Science from Lake- head University, Thunder Bay, Canada in 2022. He is currently pursuing his Master’s degree in Computer Science from the University of Ottawa.